Predicting Bitcoin Price Movements Using CatBoost and Logistic Regression with Cubic Splines

AMS 515 Project Report

Matvei Lukianov
Quantitative Finance
Stony Brook University
matvei.lukianov@stonybrook.edu

May 8, 2025

Contents

1 Introduction

High-frequency trading (HFT) in cryptocurrency markets presents unique challenges and opportunities due to the decentralized, volatile, and data-rich nature of digital assets. Among these, Bitcoin remains the most actively traded cryptocurrency, with its price movements exhibiting high sensitivity to real-time shifts in order book dynamics. Effectively modeling such rapid fluctuations requires approaches that can capture both intricate market behaviors and nonlinear relationships within the data.

This report explores the use of machine learning techniques to predict short-term Bitcoin price movements using high-frequency order book data. Specifically, we investigate two distinct methodologies: CatBoost, a gradient boosting framework well-suited for handling heterogeneous data and class imbalance, and Logistic Regression augmented with spline transformations to flexibly model nonlinear trends in the feature space. These models are evaluated not only for their predictive accuracy but also for their interpretability and computational efficiency in a real-time trading context.

By analyzing how these models perform on millisecond-level Bitcoin order book snapshots—enriched with engineered indicators such as liquidity imbalances, order flow pressure, and microstructure-derived features—this work aims to assess the practical viability of modern machine learning tools for decision-making in crypto HFT environments.

2 Problem Statement: Model Formulation

In high-frequency cryptocurrency trading, market participants rely on precise and timely predictions of price direction to inform trade execution and manage risk. Traditional statistical models often fall short in capturing the nonlinear, high-dimensional, and rapidly evolving nature of order book data. This project addresses the task of predicting imminent Bitcoin price movements by leveraging features derived from high-resolution order book snapshots.

The modeling task is framed as a binary classification problem: given a set of features at time t, predict whether the mid-price will move up, down, or remain unchanged within a short future window. The input dataset consists of millisecond-level data including bid/ask prices, volume imbalances, depth distributions, and engineered indicators reflecting short-term market pressure.

Two modeling approaches are employed:

- **CatBoost:** A gradient boosting method optimized for categorical and imbalanced data, known for its robustness and minimal feature preprocessing requirements.

- **Logistic Regression with Cubic Splines:** A more interpretable statistical model enhanced with spline-based feature transformations to capture complex relationships in a computationally efficient way.

The goal is to compare these models in terms of classification performance using metrics such as accuracy, precision, recall, and F1-score, as well as domain-specific evaluation criteria relevant to trading. Ultimately, this comparison aims to provide actionable insights into the strengths and trade-offs of tree-based versus spline-augmented linear methods for modeling high-frequency Bitcoin market behavior.

2.1 Model Objectives

This study formulates a deterministic classification framework to predict short-term Bitcoin price direction using two distinct approaches: CatBoost and Logistic Regression with spline transformations. The primary goals of this modeling effort include:

- **Directional Price Prediction:** Develop accurate classifiers capable of detecting imminent upward or downward movements in Bitcoin prices based on high-frequency order book features.
- **Feature Interpretability and Flexibility:** Compare the interpretability and expressive power of a traditional statistical model (Logistic Regression with splines) against a modern, tree-based machine learning approach (CatBoost), especially in modeling nonlinear market dynamics.
- **Real-Time Trading Suitability:** Ensure that models are computationally efficient and responsive enough to support real-time decision-making within high-frequency trading (HFT) environments.

2.2 Deterministic Model Formulation

2.2.1 Binary Classification Framework

The problem is cast as a supervised classification task, where the target label indicates the direction of the Bitcoin mid-price movement over a short future window:

$$Y = \begin{cases} 1 & \text{if price increases,} \\ 0 & \text{otherwise.} \end{cases}$$

Models are trained on a rich set of features derived from order book states, such as bid-ask spreads, volume imbalances, depth ratios, and price-action indicators.

3 Approach and Description of Solution Method

3.1 Classification Models: Theory and Intuition

This study frames Bitcoin price movement prediction as a binary classification problem, where the objective is to forecast whether the price will rise or not over a short future window. Two modeling approaches are explored:

- **Logistic Regression with Cubic Splines:** A classical yet powerful statistical model augmented with spline transformations to capture nonlinear relationships between input features and the probability of upward price movement.

- **CatBoost:** A gradient boosting algorithm optimized for handling categorical features and imbalanced datasets, capable of capturing intricate patterns in high-frequency order book data.

Both models are trained using a set of engineered features derived from high-frequency Bitcoin order book snapshots, including bid-ask spreads, volume imbalances, and short-term liquidity metrics.

3.2 Logistic Regression with Cubic Splines

Logistic regression models the log-odds of a binary outcome as a linear function of predictors. To flexibly capture nonlinearities in continuous variables while maintaining model interpretability, we apply **cubic spline transformations** to selected features. This results in a model of the form:

$$\log\left(\frac{P(Y=1 \mid X)}{1-P(Y=1 \mid X)}\right) = \beta_0 + \sum_j f_j(X_j),$$

where $f_j(X_j)$ is a smooth, nonlinear function of the predictor X_j, modeled using cubic splines.

A **cubic spline** is a piecewise-defined polynomial of degree 3. It consists of multiple cubic polynomials joined at specific data values called knots. The key idea is to construct $f_j(X_j)$ such that:

1. Each piece is a cubic polynomial on an interval between knots. 2. The full function is continuous. 3. The first and second derivatives are also continuous at each knot, ensuring smooth transitions.

Let the knots be denoted by $\xi_1 < \xi_2 < \cdots < \xi_K$. A cubic spline function with these knots can be expressed as:

$$f_j(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \sum_{k=1}^{K} \gamma_k (x - \xi_k)_+^3,$$

where $(x - \xi_k)_+^3$ is the **truncated power basis function**:

$$(x - \xi_k)_+^3 = \begin{cases} (x - \xi_k)^3 & \text{if } x > \xi_k, \\ 0 & \text{otherwise.} \end{cases}$$

The parameters α_i and γ_k are estimated from the data during model fitting. The term $(x - \xi_k)_+^3$ activates only when $x > \xi_k$, allowing local flexibility around each knot.

Example: One Spline Feature

Suppose we have a feature $x \in [0, 10]$, and we place 3 knots at $\xi_1 = 3$, $\xi_2 = 6$, and $\xi_3 = 8$. Then the spline function might look like:

$$f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \gamma_1 (x - 3)_+^3 + \gamma_2 (x - 6)_+^3 + \gamma_3 (x - 8)_+^3.$$

The model fitting process (e.g. via maximum likelihood in logistic regression) estimates all coefficients to optimize the fit.

3.3 CatBoost and Gradient Boosting

CatBoost is a high-performance gradient boosting framework based on decision trees. Unlike traditional boosting algorithms, CatBoost incorporates several innovations such as:

- Handling categorical features without one-hot encoding.
- Ordered boosting to mitigate prediction shift.
- Efficient support for imbalanced classification tasks, common in HFT environments where price remains unchanged most of the time.

The objective function for binary classification is typically based on log-loss:

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)] + \Omega(f),$$

where $\hat{p}_i$ is the predicted probability, $y_i \in \{0, 1\}$ is the true label, and $\Omega(f)$ is a regularization term to prevent overfitting by penalizing model complexity.

3.4 Gradient Boosting Process

CatBoost (and gradient boosting in general) builds an ensemble of decision trees through the following steps:

1. **Initialization:** Start with an initial prediction (e.g., a constant probability based on class distribution).

2. **Compute Gradients:** At each iteration, compute the negative gradient of the loss function with respect to the current model's output, reflecting the direction of greatest improvement.

3. **Fit a Tree:** Train a new decision tree to fit these gradients.

4. **Update Predictions:** Add the scaled output of the new tree to the model's current predictions.

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f^{(t)}(X),$$

   where η is the learning rate and $f^{(t)}(X)$ is the output of the newly trained tree. This process continues for a predefined number of iterations or until convergence.

3.5 Model Comparison and Interpretability

While CatBoost excels in capturing complex feature interactions and handling irregularities in the dataset, Logistic Regression with splines offers greater interpretability and faster inference time, which is valuable for real-time applications. This comparison allows us to weigh the benefits of modern ensemble methods against traditional statistical models in the context of crypto HFT.

4 Indicators calculation

4.1 Random Values Corresponding to Books

A classical order book is a collection of bids $(b_i, s_i^b), i = 1, ..., I^b$ and asks $(a_i, s_i^a), i = 1, ..., I^a$, where b_i and s_i^b are the bid prices and bid sizes, and similarly, a_i and s_i^a are the ask prices and ask sizes. Denote by $(\mathbf{b}, \mathbf{s^b})$ the matrix with two columns and I^b rows including bid values and sizes, and by $(\mathbf{a}, \mathbf{s^a})$ the matrix with two columns and I^a rows including ask values and sizes.

For a vector $\mathbf{x} = (x_1, ..., x_n)$, denote by $||\mathbf{x}||_1$ the L_1 norm, i.e.,

$$||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i| \, .$$

Denote by $\mathbf{w}$ a vector of non-negative weights. The vector $\mathbf{x}$ weighted by vector $\mathbf{w}$ is denoted as follows, $\mathbf{x}_w = (w_1 x_1, ..., w_n x_n)$.

For example, $\mathbf{w}$ can be denoted as vector of ones and zeros, if we want just to neglect the effect of some outlier positions or it can be calculated as exponential decay by volume, starting from some strip.

Let us denote by $\mathbf{w}^a$ and $\mathbf{w}^b$ weight vectors of ask and bid positions. We also denote by $\mathbf{a}_w$ and $\mathbf{b}_w$ the weighted ask and bid position vectors,

$$\mathbf{a}_w = (w_1^a s_1^a, ..., w_{I_a}^a s_{I_a}^a) \text{ and } \mathbf{b}_w = (w_1^b s_1^b, ..., w_{I_b}^b s_{I_b}^b) \, .$$

Let us denoted by A_w and B_w weighted discrete random values, corresponding to the ask book $\mathcal{G}^- = (\mathbf{a}, -\mathbf{s}^a)$ and bid book $\mathcal{G}^+ = (\mathbf{b}, \mathbf{s}^b)$, accordingly. These random values, A_w and B_w, have components, $\ln a_i$ and $\ln b_i$, taken with probabilities $a_w i/||a_w||_1$ for $i = 1, ..., I^a$, and $b_w i/||b_w||_1$ for $i = 1, ..., I^b$.

4.2 Indicators

4.3 Total Indicators

- Difference between volumes:

$$\frac{1}{||\mathbf{a}_w||_1 + ||\mathbf{b}_w||_1} (||\mathbf{a}_w||_1 - ||\mathbf{b}_w||_1) \, .$$

- Difference between average log prices:

$$\mathbb{E} A_w - \mathbb{E} B_w \, .$$

- Difference between volume weighted log prices:

$$\frac{1}{||\mathbf{a}_w||_1 + ||\mathbf{b}_w||_1} (||\mathbf{a}_w||_1 \, \mathbb{E} A_w - ||\mathbf{b}_w||_1 \, \mathbb{E} B_w) \, .$$

4.4 Fractional Indicators

Denote by $q_\alpha(X)$ an α quantile of the random value X. Let us divide the unit interval $(0,1)$ to N equal subintervals (α_{n-1}, α_n), i.e. $\alpha_n - \alpha_{n-1} = 1/N,\ n = 1, ...N$ and $\alpha_0 = 0,\ \alpha_N = 1$.

- Difference between stripes of log prices:

$$\int_{\alpha_{n-1}}^{\alpha_n} q_\beta(A_w)\, d\beta + \int_{\alpha_{n-1}}^{\alpha_n} q_\beta(-B_w)\, d\beta\ , \quad n = 1, ...N\ .$$

- Difference between volume weighted stripes of log prices:

$$\frac{1}{||\mathbf{a}_w||_1 + ||\mathbf{b}_w||_1} \left(||\mathbf{a}_w||_1 \int_{\alpha_{n-1}}^{\alpha_n} q_\beta(A_w)\, d\beta - ||\mathbf{b}_w||_1 \int_{\alpha_{n-1}}^{\alpha_n} q_\beta(-B_w)\, d\beta \right)\ , \quad n = 1, ...N\ .$$

4.5 Indicators based on market price estimation

In fact, the most important task is to somehow estimate market price and then by comparing it, for example, with 1% best bids and 1% best asks we can determine, where the prices would go next time. One of the simplest approaches to begin with is to take Vladimir's indicator, assume that ψ is MAE or MSE and then optimal x would be median or mean: Let us denote by X weighted discrete random values, corresponding to the whole orderbook $\mathcal{B} = (p, -s)$

- When minimizing squared differences, the mean of the order book volumes, $E(X)$, emerges as the solution:

$$x^\star = \arg\min_x \mathbb{E}[(X - x)^2] = \mathbb{E}(X)$$

This reflects that the mean minimizes the function for the corresponding squared error term.

- When minimizing absolute differences, the median of the order book volumes becomes the optimal solution:

$$x^\star = \arg\min_x \mathbb{E}[|X - x|] = \text{Median}(X)$$

This reflects that the median minimizes the function for the corresponding absolute error term.

These properties suggest that mean and median values can be utilized for their respective minimization properties depending on the choice of $\psi(x)$ in practical applications.

As a result, such indicator as the difference between such optimal x and best 1% strip of ask and bids can be used. The intuition is the following

5 Results

5.1 Dataset Description

The resulting dataset consists of 9 features, the generation of which was described in the indicators section. To calculate these factors, order book data for the BTC-USDT pair was downloaded for 1000 minutes at a 1-second frequency, resulting in 60,000 rows. Twenty percent of the data was used for the test set, with the remainder allocated to the training set. The target variable was the return 1 second ahead.

5.2 Classification metrics

Results are summarized in Table 1 below:

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
LogReg + Splines	0.646691	0.587406	0.390869	0.469395	0.670397
CatBoost	0.679197	0.633747	0.468209	0.538545	0.731145

Table 1: Classification metrics on the test set

CatBoost outperforms logistic regression with splines across all key metrics, with the largest improvement seen in recall (from 0.39 to 0.47) and F1 score. This suggests that while logistic regression with spline-transformed features captures some nonlinearity, CatBoost's boosted tree architecture more effectively learns complex decision boundaries and interactions among features.

5.3 Partial Dependence Analysis

The following figures show partial dependence plots (PDPs) comparing the predicted probabilities for both models as a function of each individual feature. Spline knots are indicated with vertical dashed lines.

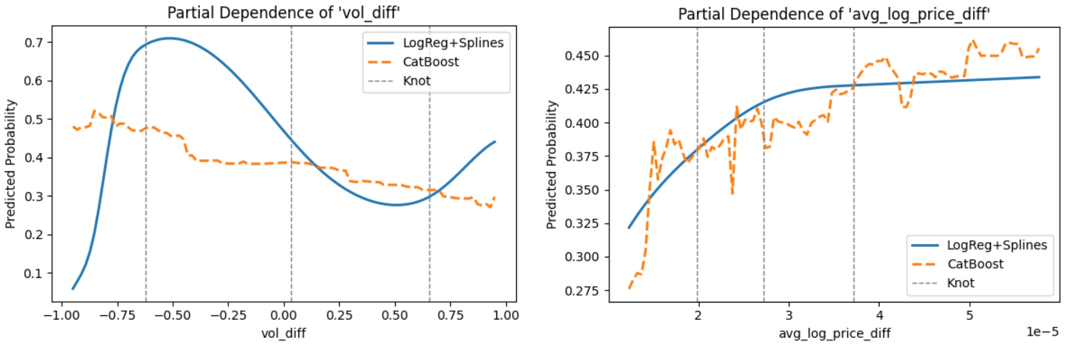

Figure 1: Partial dependence plots for `ask_median_diff` and `vol_weighted_diff_stripes`

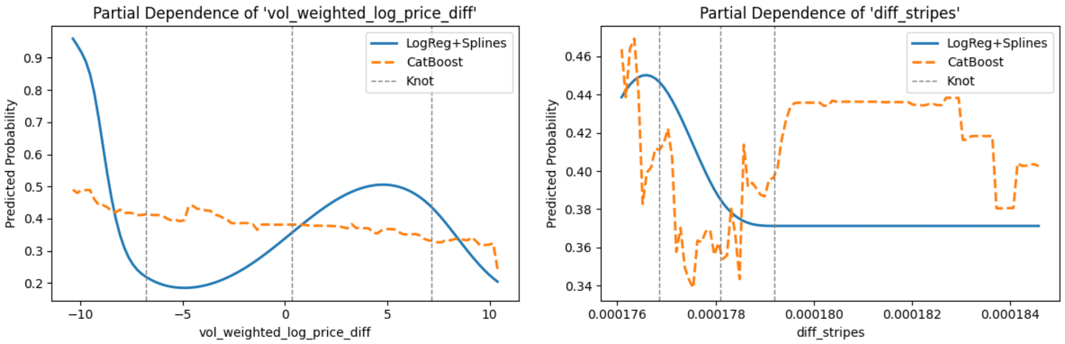

Figure 2: Partial dependence plots for `diff_stripes` and `vol_weighted_log_price_diff`

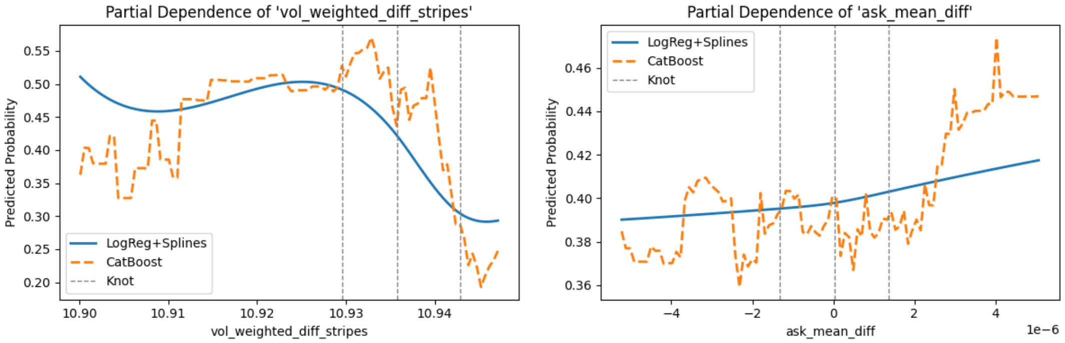

Figure 3: Partial dependence plots for `avg_log_price_diff` and `vol_diff`

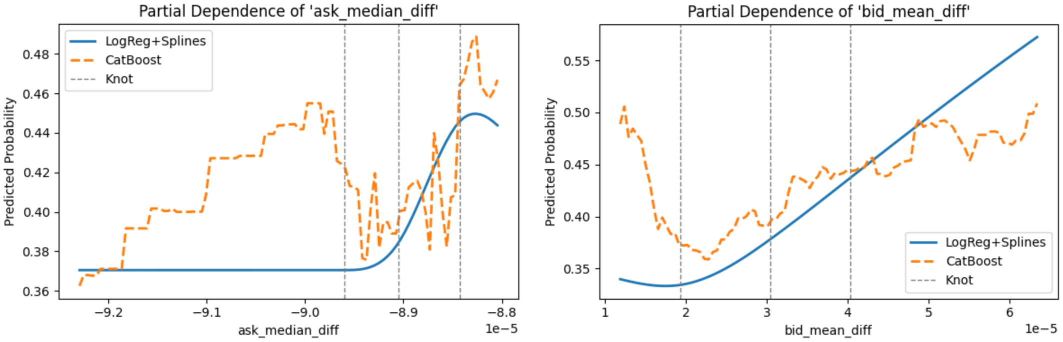

Figure 4: Partial dependence plots for `bid_median_diff` and `bid_mean_diff`

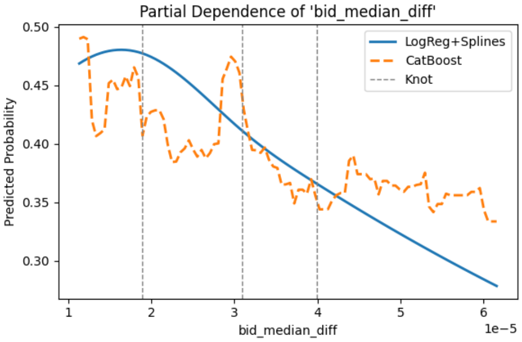

Figure 5: Partial dependence plot for `ask_mean_diff`

The partial dependence plots offer insight into how each model interprets the nonlinear effects of individual features on the probability of positive return. A key distinction arises from the structure of the models themselves: the spline-augmented logistic regression produces globally smooth and continuous transformations, while CatBoost, being an ensemble of decision trees, yields stepwise or piecewise-constant responses.

For features like `vol_weighted_log_price_diff` and `vol_diff`, the spline model captures distinct nonlinear patterns such as inflection points and plateaus. These may correspond to meaningful thresholds—points at which subtle shifts in order book pressure result in significantly altered return probabilities. In contrast, CatBoost displays more abrupt transitions, suggesting sensitivity to local discontinuities or interactions not explicitly modeled in the logistic framework.

Interestingly, features such as `avg_log_price_diff` and `bid_mean_diff` show relatively clean monotonic or convex effects under CatBoost, even more so than under the spline model. This hints that CatBoost may be clustering data into regimes or microstructures where signal strength varies. These differences may reflect the underlying data-generating processes: e.g., momentum-like behavior versus mean-reverting shocks.

We can loosely cluster features based on the behavior observed across both models:

- **Directional/momentum features** (e.g., `avg_log_price_diff`, `ask_median_diff`) exhibit strong monotonic patterns and contribute significantly to both models.
- **Imbalance indicators** (e.g., `bid_mean_diff`, `vol_diff`) show more complex or regime-sensitive behavior, which CatBoost models more adaptively.
- **Noisy or low-signal features** (e.g., `ask_mean_diff`) produce flat or unstable patterns, suggesting weaker predictive utility.

One important observation is that the spline model, despite being more interpretable, may underfit in regions where local variation is high, missing sharp transitions that are crucial for fast-moving financial dynamics. Conversely, CatBoost may overfit to noise in low-signal features or during rare market events, though its built-in regularization and validation criteria help mitigate this.

Overall, these plots emphasize that while spline-transformed logistic regression provides a global, interpretable lens on feature behavior, CatBoost is capable of capturing richer, localized interactions that are essential for predictive performance in high-frequency financial contexts. These differences make a strong case for hybrid approaches: for instance, using spline models for feature screening or interpretability, and tree-based models for execution-layer prediction.

5.4 Model Evaluation and Diagnostics

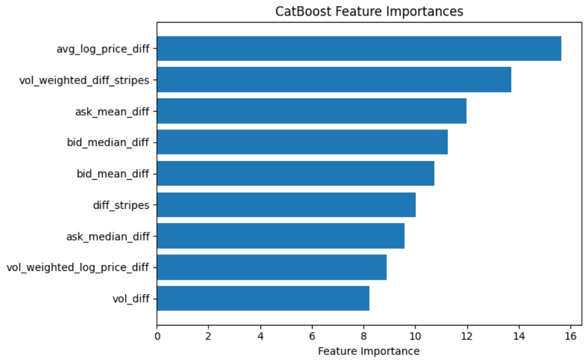

Figure 6: Feature importances from the CatBoost model

Fig. 6 shows that `avg_log_price_diff` and `vol_weighted_diff_stripes` are the most influential features according to CatBoost. These variables likely reflect directional flow and short-term price imbalances. Other features such as `ask_mean_diff` and `bid_mean_diff` still contribute meaningfully, but with less weight. The feature ranking aligns well with intuition and is consistent with partial dependence findings.

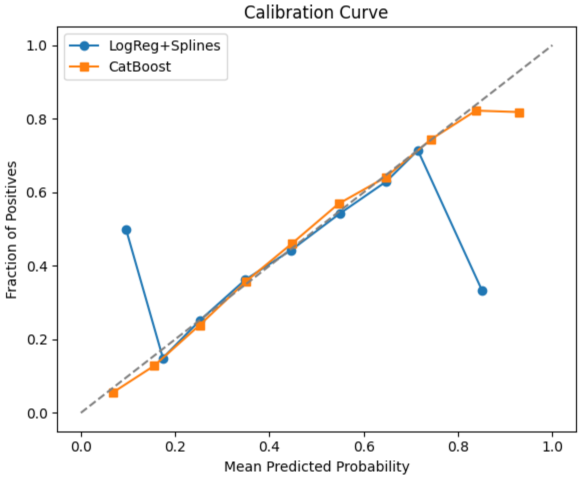

Figure 7: Calibration curves for LogReg+Splines and CatBoost

The calibration curve in Fig. 7 demonstrates that CatBoost produces probabilities that are better aligned with actual outcome frequencies. The spline model tends to overpredict in higher probability bins, leading to poor calibration for confident predictions. This is critical in risk-sensitive trading systems, where reliability of confidence matters.

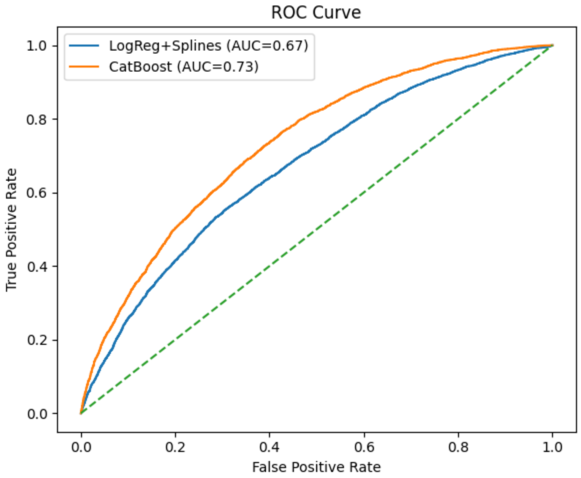

Figure 8: ROC curves with AUC scores

Fig. 8 compares the ROC curves for both models. CatBoost achieves a higher AUC (0.73) compared to the spline model (0.67), showing superior ability to rank true positives above negatives across thresholds. This is expected due to the ensemble model's flexibility and depth.

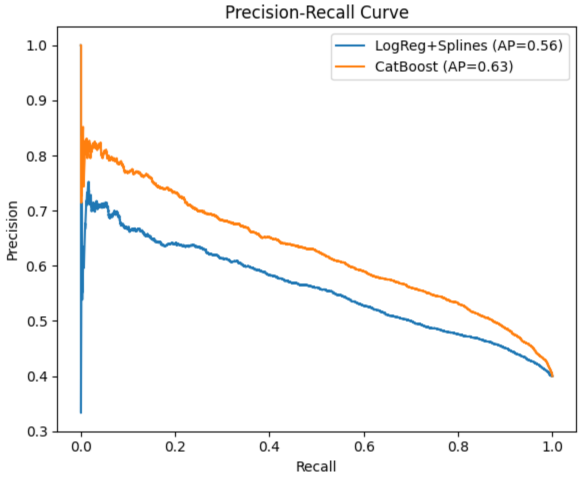

Figure 9: Precision-Recall curves with average precision

In Fig. 9, CatBoost also demonstrates higher average precision (AP = 0.63) compared to LogReg+Splines (AP = 0.56), indicating it is more effective under class imbalance. This reinforces CatBoost's advantage in high-frequency financial classification tasks.

6 Conclusions

In this study, we explored the use of logistic regression with cubic splines and CatBoost for predicting short-term return direction in high-frequency cryptocurrency markets. Our modeling was based on engineered limit order book features from BTC-USDT data sampled at a 1-second frequency.

The spline-augmented logistic regression model provided smooth and interpretable approximations of nonlinear relationships, revealing key market dynamics through partial dependence plots. However, despite its transparency, the model was outperformed by CatBoost in almost every evaluation metric.

CatBoost achieved higher accuracy, recall, F1 score, and ROC AUC, confirming its superior classification capability in complex, noisy environments like high-frequency trading. Moreover, its calibration and precision-recall behavior proved more reliable under class imbalance—a crucial consideration in practical trading systems.

Feature importance rankings from CatBoost aligned well with domain intuition and were supported by spline-based visual analysis. This consistency across models enhances confidence in the extracted signals.

Overall, while logistic regression with splines remains valuable for diagnostic and explanatory purposes, gradient-boosted trees such as CatBoost are more effective for production-level signal prediction. Future work may focus on ensemble combinations, deeper temporal modeling, or extending these approaches to multi-asset portfolios.